AutomaticallyMountPartitions

Mounting Partitions Automatically

There are broadly two aproaches -

- Per-user mounting (usually under /media)
- Systemwide mounting (anywhere, often under /mnt)

Per-user mounting does not require root access, it's just automating the desktop interface. Systemwide mounts (/etc/fstab) can allow access from before login, and are therefore much more suitable for access through a network, or by system services.

Commands should be entered on a terminal (Type **terminal** in the program launcher of recent unity based Ubuntu releases, or select Applications -> Accessories -> Terminal from the menus on older releases).

Per-User Mounts

udisks

This is the modern replacement for **gnome-mount**. It's not gnome specific.

When you mount a disc normally with the file browser (nautilus etc) it mounts disks by interacting with udisks behind the scenes.

You can do the same thing on the command line with the udisks tool.

eg

```
/usr/bin/udisks --mount /dev/sdb1
```

The bit after **--mount** is the device name of the partition you want to mount. (**/dev/something**)

Finding the device name of your Partition

- Open your partition in nautilus (this makes sure it's mounted)
- type **mount** in a terminal. You should see a line with your disk name on it like:

```
/dev/sdb1 on /media/My-Happy-Disk type vfat ....
```

Finding the UUID of your partition

A device name like /dev/sdb1 is based on where your physical drive is plugged in and the order the drives were made available to the computer, so if your computer changes the same command could mount a different partition. It's possible for this to happen just from a software upgrade.

The solution is to use a UUID. A UUID is a globally unique name for the partition. A UUID will remain the same if you put an internal disk into an external USB caddy, or change the name of the partition.

- Type **ls -al /dev/disk/by-uuid/** you will see an entry that matches the name you saw before:

```
lrwxrwxrwx 1 root root  10 2012-02-15 10:23 1313-F422 -> ../../sdb1
```

The UUID in the example is *1313-F422*

You can now determine the command you need for mounting the device by UUID. For our example it would be

```
/usr/bin/udisks --mount /dev/disk/by-uuid/1313-F422
```

Of course you need to replace *1313-F422* with the UUID of the device you want to mount.

Adding to startup

- From the Ubuntu dash (click logo in top left) find **startup applications** or press Alt+F2 and type **gnome-session-properties**
- Push the **Add** button.
- Choose a name, paste in your command and push the **Add** button

Gnome-Mount

⛔ This method is only available in Hardy and Karmic.

There is no need to setup mountpoints or filesystems. This is particularly interesting if you want to use the automatically created mountpoints instead of manually specifying them for each disk.

Installation

```
sudo apt-get install gnome-mount
```

Usage

You can mount a disk as a normal user via

```
gnome-mount -p myDiskLabel
```

where myDiskLabel is your disk name (e.g Data). To have the disk mounted each time you startup (thus removing the password prompt on first usage of the disk), simply add above line to your list of startup applications (System->Preferences->Startup Applications). Note that while mounting through gnome-mount will not need a password, unmounting or remounting via Nautilus will still invoke the password prompt.

Systemwide Mounts

Three methods will be discussed:

1. The first method is manually editing Ubuntu's filesystem table. This sounds more complex than it really is.
2. The second method, for versions 6.06 and later, is described at MountingWindowsPartitions.
3. The third - simple - method is to install the **pysdm** package (in Gutsy) and then use **System-Administration-Storage Device Manager**without any manual editing of the fstab file, and disregard most of the instructions that follow. (NOTE: psydm removed from repositories in 12.10 and presumably beyond - see http://ubuntuforums.org/showthread.php?p=12338212. Use of arios automount or mount manager not recommended)

Manual Setup Help

To mount hard disk partitions, you should have a basic understanding of the information below.

Once installed you can browse to System > Network and File Systems >

Basic understanding is still required...

Viewing the system's physical information

To read the layout of the physical disks in the system, the 'fdisk' command is used. Before panicking, realize that fdisk will be used with only non-destructive options; specifically, it will be used with 'l' (lower-case 'L', not '1'), which lists the partition table of the specified disk.

```
sudo fdisk -l
```

fdisk will print the partition tables of all your disks. The output from this command will look something like this;

```
Disk /dev/hda: 40.0 GB, 40013261856 bytes
255 heads, 63 sectors/track, 4439 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start          End      Blocks    Id   System
/dev/hda1 *         1              638      5124703   b    W95 FAT32
/dev/hda2           639            4525     31222327  83   Linux
/dev/hda3           4526           4635     497980    82   Linux swap
```

If the system has multiple hard disk drives, multiple lists will be displayed.

Deciding which partitions to mount

Most systems only have */dev/hda*, which is the hard disk drive, and */dev/hdc*, which is the CD-ROM, or optical, drive. If more were listed when the command above was run, they can be identified as follows: *hda* is the first drive on the first IDE channel (0:0), *hdb* is the second drive (0:1), *hdc* is the third drive (1:0), and *hdd* is the fourth (1:1). SCSI and S-ATA disks have names like *sda* and *sdb*.

Look through the list generated above to identify the partition(s) to be mounted. The following table lists some common 'System' types, which may help this process.

System name	English name	Linux type
W95 FAT32	Microsoft FAT32	vfat
W95 FAT32 (LBA)	Microsoft FAT32	vfat
W95 FAT16 (LBA)	Microsoft FAT16	vfat
W95 Ext'd (LBA)	Microsoft extended partition	Not used
NTFS volume set	Microsoft NTFS	ntfs
NTFS volume set	Microsoft NTFS with read-write access	ntfs-3g
Apple_HFS	Apple HFS	hfsplus

A list of the form '/dev/hda1: /media/windows/ (vfat)', where '/dev/hda1' is the device, '/media/windows' is the arbitrary location where the partition will appear when mounted, and 'vfat' is the Linux type, should be created on paper, containing all partitions to be added.

Preparing the system

Look through the list which was just created. For every location ('/media/windows'), run the following command.

```
ls /media/windows
```

If a response like *ls: /media/windows: No such file of directory* is returned, the location is open. If a list of files or nothing is returned, the location exists already, and the planned location will need to be altered.

Once every location has been confirmed as free, run the following command for each entry, replacing '/media/windows' with the chosen location.

```
sudo mkdir /media/windows
```

Editing Ubuntu's filesystem table

It is possible to break Ubuntu if some of the earlier lines in the file opened during this step are modified, so be sure to read this section carefully.

Ubuntu's filesystem table is located at '/etc/fstab'. Open this file for editing by running the following command for Ubuntu

```
gksu gedit /etc/fstab
```

or this command for Kubuntu

```
kdesu kate /etc/fstab
```

or command line

```
nano -w /etc/fstab
```

The file opened contains lines of the form *<device> <location> <Linux type> <options> <dump> <pass>*. Every element in this line is separated by whitespace (spaces and tabs).

```
# /etc/fstab: static file system information.
#
# <file system>                            <mount point>       <type>            <options>
<dump>    <pass>
proc                                       /proc               proc              defaults
0         0
# /dev/hdb1
UUID=e776c7ac-61d1-4ca2-8b59-0c874933b2de  /                   ext3
defaults,errors=remount-ro  0         1
# /dev/hdb5
UUID=5ff60812-91c2-42bd-b628-8cb530f073fb  none                swap              sw
0         0
/dev/hdc                                   /media/cdrom0       udf,iso9660
user,noauto,exec            0         0
/dev/fd0                                   /media/floppy0      auto
rw,user,noauto,exec         0         0
```

```
/dev/mapper/sil_aiaedhaeafaa1              /media/raid         reiserfs
user,nosuid,exec,nodev      0         0
```

<file system>

The first field, (fs_spec), describes the block special device or remote filesystem to be mounted.

For ordinary mounts it will hold (a link to) a block special device node (as created by mknod(8)) for the device to be mounted, like `/dev/cdrom` or`/dev/sdb7`. For NFS mounts one will have <host>:<dir>, e.g., `knuth.aeb.nl:/`. For procfs, use proc.

Instead of giving the device explicitly, one may indicate the (ext2 or xfs) filesystem that is to be mounted by its UUID or volume label (cf. e2label(8) or xfs_admin(8)), writing LABEL=<label> or UUID=<uuid>, e.g., LABEL=Boot or `UUID=3e6be9de-8139-11d1-9106-a43f08d823a6`. This will make the system more robust: adding or removing a SCSI disk changes the disk device name but not the filesystem volume label. The UUID of a drive can be determined by typing `ls -l /dev/disk/by-uuid/` in a shell.

<mount point>

The second field, (fs_file), describes the mount point for the filesystem. For swap partitions, this field should be specified as `none`. If the name of the mount point contains spaces these can be escaped as `\040`.

<type>

The third field, (fs_vfstype), describes the type of the filesystem. Linux supports lots of filesystem types, such as adfs, affs, autofs, coda, coherent, cramfs, devpts, efs, ext2, ext3, hfs, hpfs, iso9660, jfs, minix, msdos, ncpfs, nfs, ntfs, proc, qnx4, reiserfs, romfs, smbfs, sysv, tmpfs, udf, ufs, umsdos, vfat, xenix, xfs, and possibly others. For more details, see mount(8). For the filesystems currently supported by the running kernel, see /proc/filesystems. An entry swap denotes a file or partition to be used for swapping, cf. swapon(8). An entry ignore causes the line to be ignored. This is useful to show disk partitions which are currently unused.

<options>

The fourth field, (fs_mntops), describes the mount options associated with the filesystem.

It is formatted as a comma separated list of options. It contains at least the type of mount plus any additional options appropriate to the filesystem type. For documentation on the available options for non-nfs file systems, see mount(8). For documention on all nfs-specific options have a look at nfs(5). Common for all types of file system are the options `noauto` (do not mount when "mount -a" is given, e.g., at boot time), `user` (allow a user to mount), and `owner` (allow device owner to mount), and `comment` (e.g., for use by fstab-maintaining programs). The `owner` and `comment`options are Linux-specific. For more details, see mount(8).

<dump>

The fifth field, (fs_freq), is used for these filesystems by the dump(8) command to determine which filesystems need to be dumped. If the fifth field is not present, a value of zero is returned and dump will assume that the filesystem does not need to be dumped.

<pass>

The sixth field, (fs_passno), is used by the fsck(8) program to determine the order in which filesystem checks are done at reboot time. The root filesystem should be specified with a fs_passno of 1, and other filesystems should have a fs_passno of 2. Filesystems within a drive will be checked sequentially, but filesystems on different drives will be checked at the same time to utilize parallelism available in the hardware. If the sixth field is not present or zero, a value of zero is returned and fsck will assume that the filesystem does not need to be checked.

<option> example

To learn more about `options`, type 'man mount'.

Description	Accessible by everyone	Accessible by a subset of users**
FAT(16/32) partition	user,auto,fmask=0111,dmask=0000	user,auto,fmask=0177,dmask=0077,uid=1000
NTFS partition*	rw,auto,user,fmask=0111,dmask=0000	rw,user,auto,fmask=0177,dmask=0077,uid=1000
Apple Partition	user,auto,file_umask=0111,dir_umask=0000	user,auto,file_umask=0177,dir_umask=0077,uid=1000

*If you want write access to your file system, you should set the filesystem type to 'ntfs-3g' instead of 'ntfs'. You may need to install the package 'ntfs-3g' for this to work, so make sure it is installed **before** you use ntfs-3g.

**uid=1000 restricts access to the user created while installing Ubuntu. 1001 is the user created after that, and so forth. gid=# may be used with or in place of uid to grant access to a group. However, group and user enumeration is beyond the scope of this article.

<type> example

Note for international users: if your filesystem contains funny symbols, you may need to add an option for utf-8 support.

Filesystem type	Option to enable utf-8 support
ntfs	nls=utf8
vfat	utf8
smbfs	iocharset=utf8,codepage=unicode,unicode

fstab example

So, to grant all users access to '/dev/hda1', which will be located at '/media/windows', and is of type 'vfat', the line added would be.

```
/dev/hda1   /media/windows   vfat   user,fmask=0111,dmask=0000   0   0
```

This entry is case-sensitive. In general, lower-case letters are used to avoid confusion.

'dump' and 'pass' are only of use to native filesystems. They can be set to '0' for all additional partitions.

For every item in the list of partitions to be mounted, add one line of the form above to the end of the fstab file.

Mounting Fakeraid

Its assumed you have formated you raid set using `dmraid` command with instructions found at FakeRaidHowto

You need to know the name of your raid set

```
sudo dmraid -ay
```

```
RAID set "sil_aiaedhaeafaa" already active
RAID set "sil_aiaedhaeafaa1" already active
```

Edit `fstab` as per instructions above :

```
nano -w /etc/fstab
```

An example line to add

```
/dev/mapper/sil_aiaedhaeafaa1  /media/raid  reiserfs  user,nosuid,exec,nodev  0  0
```

Make sure you create the directory `/media/raid`

```
mkdir /media/raid
```

Reboot

Mounting and checking the partitions

In the terminal, type the following command.

```
sudo mount -a
```

To verify that the partitions were mounted properly, open Gnome's file browser and direct it to the locations at which the partitions were mounted. Click the 'File System' button to access '/', and navigate from there. If the partition being examined contains files, the modifications were successful, and the partitions will be automatically mounted every time the system is restarted.

If no files are found, please see XChatHowto and join #ubuntu on irc.freenode.net.

Hints, Tips, and Technical Information

Adding a bookmark

Bookmarks may be added to help speed access to commonly used files and locations, such as mounted partitions.

While browsing the filesystem using Gnome's file browser, click 'Bookmarks', then 'Add Bookmark'. A new entry will be placed in the bar on the left, and the location will appear under the 'Places menu'.

Alternatively, bookmarks may be added while opening or saving a file by clicking the 'Add' button.

To remove a bookmark, click either 'Edit Bookmarks', or the 'Remove' button, either of which is the counterpart of the two addition methods stated above.

More technical tips

Symlinking for greater convenience

If navigating to a partition's mount point seems inconvenient, even with the links on the left of Gnome's file browser, a link can be placed on the desktop, or anywhere else, for that matter.

Try the following command.

```
ln -s /media/windows ~/Desktop/
```

A link to the directory '/media/windows' will be placed on the desktop. Files may be dragged into it, it may be opened, it can be renamed and moved, and if it proves to be annoying, it can be deleted like any other file without risking damage to its contents.

This process is called symlinking because the link created is symbolic. It merely points to the location being referenced.

Unmounting a partition to prevent unwanted access

While it's easy and effective to set permissions on partitions, there may be times when setting permissions won't be enough. Sometimes, people who can't be trusted with important information, such as a term paper, may need access to a system, and modifying the filesystem table may be impractical. When things like this happen, it's easy to archive data to preserve permissions and stick it on a mounted partition or device, then unmount the device, preventing "accidental" access. (The potentially jokingly malicious user would need to know the filesystem table, and while that information isn't well hidden, it's hard to access without drawing attention.)

Try the following command to unmount a partition.

```
sudo umount /media/windows
```

The partition mounted at '/media/windows' will be unmounted, and attempts to access it will yeild only an empty directory. To bring it back later, either reboot, or simply run the following command.

```
sudo mount /media/windows
```